

## WEST Search History





DATE: Sunday, January 25, 2004

Hide?	<u>Set Name</u>	<u>Query</u>	<u>Hit Count</u>
		<i>DB=USPT; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L6	(rpc or (remot\$ adj procedur\$ adj call\$)) same ((delay\$ or defer\$ or laten\$ near4 (rebuild\$ or reconstruct\$ or build\$ or construct\$))	7
<input type="checkbox"/>	L5	(rpc or (remot\$ adj procedur\$ adj call\$)) same (object near4 (rebuild\$ or reconstruct\$ or build\$ or construct\$))	65
<input type="checkbox"/>	L4	719/330[ccls]	184
<input type="checkbox"/>	L3	709/330[ccls]	0
<input type="checkbox"/>	L2	(rpc or (remot\$ adj procedur\$ adj call\$)) same ((defer\$ or delay\$ or laten\$ near4 (rebuild\$ or reconstruct\$))	3
<input type="checkbox"/>	L1	(rpc or (remot\$ adj procedur\$ adj call\$)) near12 (rebuild\$ or reconstruct\$)	13

END OF SEARCH HISTORY



US006408342B1

(12) **United States Patent**  
**Moore et al.**

(10) **Patent No.: US 6,408,342 B1**  
 (45) **Date of Patent: Jun. 18, 2002**

(54) **COMMUNICATIONS FRAMEWORK FOR SUPPORTING MULTIPLE SIMULTANEOUS COMMUNICATIONS PROTOCOLS IN A DISTRIBUTED OBJECT ENVIRONMENT**

(76) **Inventors:** Keith E. Moore, 3090 Mauricia Ave., Santa Clara, CA (US) 95051; Evan Kirshenbaum, 441 Bella Corte, Mountain View, CA (US) 94043

(\*) **Notice:** Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 42 days.

(21) **Appl. No.: 08/827,213**

(22) **Filed: Mar. 28, 1997**

(51) **Int. Cl.<sup>7</sup> ..... G06F 9/46**

(52) **U.S. Cl. .... 709/330; 709/315**

(58) **Field of Search .... 709/300, 301, 709/303, 330, 328, 315, 316**

(56) **References Cited**

#### U.S. PATENT DOCUMENTS

5,218,699 A *	6/1993	Brandle et al. ....	395/650
5,307,490 A *	4/1994	Davidson et al. ....	395/650
5,499,343 A *	3/1996	Pettus ....	395/200.2
5,511,197 A *	4/1996	Hill et al. ....	395/700
5,526,491 A *	6/1996	Wei ....	395/200.09
5,539,909 A *	7/1996	Tanaka et al. ....	395/700
5,566,302 A *	10/1996	Khalidi et al. ....	395/200.9
5,758,186 A *	5/1998	Hamilton et al. ....	709/200
5,822,521 A *	10/1998	Gartner et al. ....	395/705
5,875,335 A *	2/1999	Beard ....	395/705
5,887,172 A *	3/1999	Vasudevan et al. ....	395/684

#### OTHER PUBLICATIONS

Grady Booch, Object Oriented Design with Applications, 1991, Benjamin/Cummings Publishing Company, p. 76-114.\*

A. Birrell, et al, "Network Objects", ACM, pp. 217-230, Aug. 1993.\*

H. Carr, et al, "Compiling Distributed C++", IEEE, pp. 496-503, 1993.\*

A. Dave, et al, "Proxies, Application Interfaces, and Distributed Systems", IEEE, pp. 212-219, 1992.\*

C. Enright, et al, "An Object Re-Engineering of the Remote Procedure Call, Streams and Transport Layer Interface", IEEE, pp. 602-605, 1995.\*

\* cited by examiner

**Primary Examiner—Majid Banankhah**

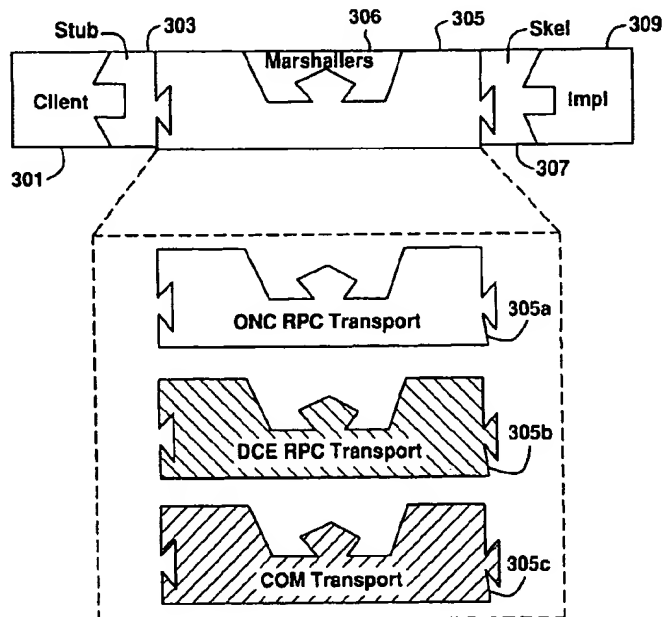
**Assistant Examiner—Sue Lao**

#### (57) **ABSTRACT**

A communication framework supporting multiple communications protocols. The communications framework having a remote procedure call class providing an interface for an apply method, the apply method referencing a remote object, an operation to be performed, and an argument list. The communications framework also having at least one remote procedure call transport deriving from the remote procedure call class, each remote procedure call transport providing an implementation for the apply method whose interface is provided by the remote procedure call class.

Other systems and methods are disclosed.

**30 Claims, 14 Drawing Sheets**



[First Hit](#)   [Fwd Refs](#)

Generate Collection

Print

L6: Entry 3 of 7

File: USPT

Jun 18, 2002

DOCUMENT-IDENTIFIER: US 6408342 B1

TITLE: Communications framework for supporting multiple simultaneous communications protocols in a distributed object environment

Detailed Description Text (236):

The communications framework creates a new ObjectReference 501 for a target object whenever a target object is first registered with the communications framework. Optionally, the construction of the ObjectReference 501 may be delayed until it is needed, thus avoiding any unnecessary ObjectReference 501 creation. The created ObjectReference 501 is passed to other processes either by returning the ObjectReference 501 as a return parameter from a remote procedure call to another process, or by passing the ObjectReference 501 as a parameter in an outbound remote procedure call. Alternatively, the ObjectReference 501 can be made known to other processes by placing it in a shared medium, such as a shared disk file.

9. 2001



US006189046B1

(12) **United States Patent**  
**Moore et al.**

(10) Patent No.: **US 6,189,046 B1**  
 (45) Date of Patent: **\*Feb. 13, 2001**

(54) **MECHANISM AND METHOD FOR  
 MERGING CACHED LOCATION  
 INFORMATION IN A DISTRIBUTED OBJECT  
 ENVIRONMENT**

5,758,186 \* 5/1998 Hamilton et al. .... 395/831  
 5,802,590 \* 9/1998 Draves ..... 711/164  
 5,892,910 \* 4/1999 Safadi ..... 395/200.47

#### OTHER PUBLICATIONS

(75) Inventors: **Keith E. Moore**, Santa Clara; **Evan  
 Kirshenbaum**, Mountain View, both of  
 CA (US)

(73) Assignee: **Hewlett-Packard Company**, Palo Alto,  
 CA (US)

(\*) Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

Under 35 U.S.C. 154(b), the term of this patent shall be extended for 0 days.

(21) Appl. No.: **08/828,027**

(22) Filed: **Mar. 27, 1997**

(51) Int. Cl.<sup>7</sup> ..... **G06F 9/54**

(52) U.S. Cl. .... **709/315; 709/217**

(58) Field of Search ..... **395/683; 709/304,  
 709/303, 315, 330, 310, 217, 228**

(56) **References Cited**

#### U.S. PATENT DOCUMENTS

5,136,716 \* 8/1992 Harvey et al. .... 395/800  
 5,291,593 \* 3/1994 Abraham et al. .... 395/600  
 5,539,909 \* 7/1996 Tanaka et al. .... 395/700  
 5,724,588 \* 3/1998 Hill et al. .... 395/684  
 5,737,607 \* 4/1998 Hamilton et al. .... 395/701

(Chappell) David Chappell. "Understanding ActiveX and OLE" p. 51-52, Sep. 17, 1996.\*

Birrell, Andrew et al. "Network Objects" p. 219-221, Dec. 1993.\*

BNR Europe Limited. "OMG Object Request Broker 2.0 Interoperability and Initialisation RFP Response". p. 1-29, Mar. 4, 1994.\*

ICL submission to the OMG Object Request Broker 2.0. "ORB Interoperability", p. 7-14, Mar. 7, 1994.\*

\* cited by examiner

Primary Examiner—Alvin E. Oberley

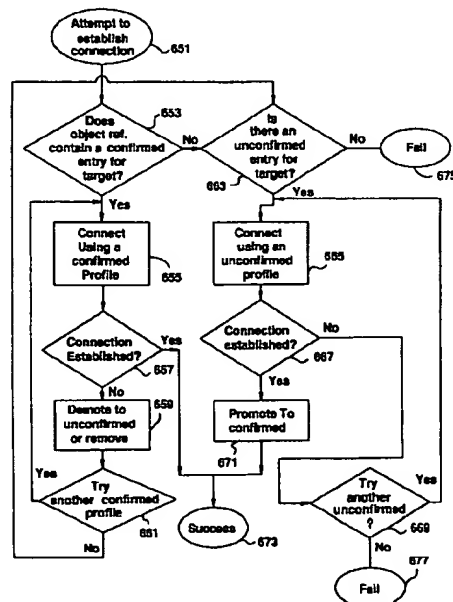
Assistant Examiner—Lewis A. Bullock, Jr.

(57) **ABSTRACT**

In a method of operating a computer system having a plurality of processes, creating a plurality of object references, each object reference corresponding to a target object. The processes optionally executing on a plurality of computers connected by a network. For each object reference, creating a table of binding information hints. The table is indexed by a particular transport protocol and each entry in the table of binding information hints includes information to be used to attempt to establish a connection from the process to the target object using the indexing transport protocol. Merging the tables of binding information hints upon receiving an object reference.

Other systems and methods are disclosed.

**13 Claims, 14 Drawing Sheets**



[First Hit](#)   [Fwd Refs](#)**End of Result Set**

Generate Collection

Print

L6: Entry 7 of 7

File: USPT

Feb 13, 2001

DOCUMENT-IDENTIFIER: US 6189046 B1

TITLE: Mechanism and method for merging cached location information in a distributed object environment

Detailed Description Text (164):

The communications framework creates a new ObjectReference 501 for a target object whenever a target object is first registered with the communications framework. Optionally, the construction of the ObjectReference 501 may be delayed until it is needed, thus avoiding any unnecessary ObjectReference 501 creation. The created ObjectReference 501 is passed to other processes either by returning the ObjectReference 501 as a return parameter from a remote procedure call to another process, or by passing the ObjectReference 501 as a parameter in an outbound remote procedure call. Alternatively, the ObjectReference 501 can be made known to other processes by placing it in a shared medium, such as a shared disk file.